ECE 176 Final Project: Self-Supervised Image Colorization

Patrick Youssef
University of California, San Diego
psyoussef@ucsd.edu

Abstract

We'll go over some of the modern techniques to colorize grayscale images using Convolutional Neural Networks (CNNs). After reviewing some recent work, we will implement one of the mentioned models and present the results. In particular, we will implement (*1*) for their work in deep learning based image colorization. Their use of a pretrained feature detector based on an Inception-ResNet was very interesting and would help when working with limited training resources. We'll go over my particular implementation and the results on the Places365 dataset near the end of this paper.

1 Introduction

The utility of coloring grayscale images have could have a strong impact on the historical community. The ability to generate the first colorized versions of images or revitalize and remaster old content is a great pursuit. It is obvious that a possible solution from the deep learning toolkit are CNNs. Although we have many tools and references at our disposal, the problem is still a challenging one as fooling human visual understanding is a monumental task.

Given the prior, I will implement a model based on the works done at the KTH Royal Institute of Technology and train it on a new dataset to see how well it performs. An aspect of the model that was of particular interest was the pretrained feature decomposition which will be a great exercise to implement in PyTorch.

Due to time constraints, I was only able to train for a limited dataset which penalizes the ability for the model to generalize well.

1.1 Organization

The rest of the paper will continue off of this focusing first on the related and current works in section 2, then my implementation and the method I took in section 3, ending with my results in section 4.

2 Related Works

As mentioned briefly in the abstract we will later implement the model described in (*1*) but I also wanted to mention a prior and well-cited work that I actually attempted before this model. The work by Zhang et. al. is one of the most cited in the world of image colorization (*2*). We will cover some of the differences and the context of the problem moving forward.

2.1 The Loss Problem

One of the primary issues with the majority of colorization models is the apparent ambiguity of colorizing many objects we are familiar with. As an example consider the table 1 detailing possible

Table 1: Possible Object Colors

Item	Possible Colors
Grass	Green, Brown, Yellow
Apple	Green, Red, Yellow
Shirt	Green, Red, Black, Yellow, Brown, etc.
Sky	Mostly Blue, Sometimes Orange

colors for, in grayscale, visually similar items. It's obvious to think about the ambiguity of colors in objects, especially when considering our desire to colorize and pop as a society.

This raises an issue where typical loss function will force the model to average the colors of the representation of objects. For objects and features that don't change very much this is ok, the sky is a good example, as the average color lies close to our expectation. For most other objects this leads to a strong desaturation leaving most images gray and sepia.

The work done by Zhang et. al. has made a massive step towards a better solution by treating the problem as a clasffication problem rather than a regression problem. Essentially, their work is in creating a custom loss that bins the color space and looks at the possible colors any particular object can take and look at losses against that as opposed to a regression type loss that would force the aforementioned behavior.

2.2 Why Something Else

Although I just mentioned how great this new solution is, I had a very difficult time getting the loss function to work but also did not enjoy the sequential structure of the model. Seeing how well skipped connection could help in visual models from Hwk4, I decided to find an alternate model and landed with the work from KTH.

3 Method

As mentioned before I opted to implement the model by KTH (*1*) as I liked their usage of skipped connection and pretrained models. I think it would he helpful to first formalize the problem we are trying to solve.

3.1 Problem Statement

We are looking to generate two new channels for a single channel input image of size HxW. Although many color spaces exist, the CIE L*a*b* color space (*3*) is well suited to this problem as it's very easily separable into our inputs and outputs with $L*$ being the grayscale luminance input and $a * b*$ being our output colored components. We can then layer the input with the output to form a colorize image in the Lab color space. Generally speaking our model is a function such that:

$$\mathbf{f}(I_L) \rightarrow (I_a, I_b)$$

This is a great problem statement compared to other visual learning problems as it can be formulated as a self-supervised model. Given an RGB input image we can convert the image into the Lab color space and separate into our input and labels. That being said, dataset requirements are merely having RGB colorized images.

3.2 Architecture

As mentioned prior, the architecture of this network is one that utilizes pretrained models and skipped connections. You can see this in Figure 3.2. The model can be separated into 4 distinct components.

- Encoder: Converts input image into a feature set
- Feature Extractor: Generates additional features from pretrained model

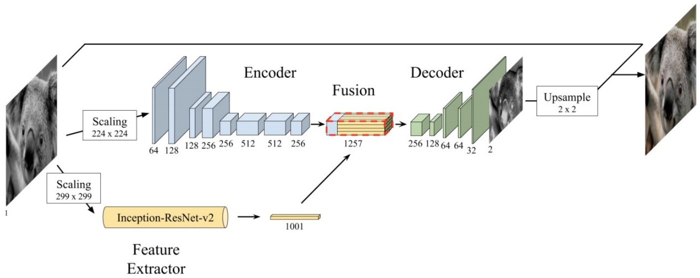

Figure 1: Model Architecture (*1*)

Table 2: Possible Object Colors

Parameter	Value
Epochs	70
Batch Size	40
Learning Rate	0.0012
Weight Decay	1e-6

- Fusion: Fuses encoder features and extracted features
- Decoder: Deconvolves features back into the required size

The exact details on their architecture are better detailed in their paper and within my code, what I will mention here is that each convolution is followed by a ReLu and Batchnorm based on my experiences in the course. Kernel sizes are globally small at 3 for most of the network steps.

4 Experiments

Here I detail my implementation, training, and results given my readings of the papers and the work I have done.

4.1 Dataset Used

Given the loose requirements on the dataset as this is a self-supervised problem, I opted to use the MIT Places365 dataset to test (*4*). I chose this dataset as I believed it would be a decent parallel to the type of content we are looking to colorize. Other datasets such as ImageNET are much more scoped in on what the images represent whereas Places365 shows more of a typical human viewport that would be expressed in historic imagery and classic content.

4.2 Training

For training the model I chose to use the Adam optimizer and trained over 1000 images. For the results shown I used the hyperparameters in Table 2.

Below in Figure 4.2 also the training loss curve over minibatches, given the limited GPU memory I was not able to run both validation and training during the training process so I only have losses for the training set. Due to this my model is likely overfitted but this is something I cannot validate until later.

4.3 Results

Here is where I will detail the results of the model and where it performed well and where some ares of improvement need to be made. Given the small dataset, I do not believe that the model will generalize well to new images. To try to characterize this difference, I present both samples from

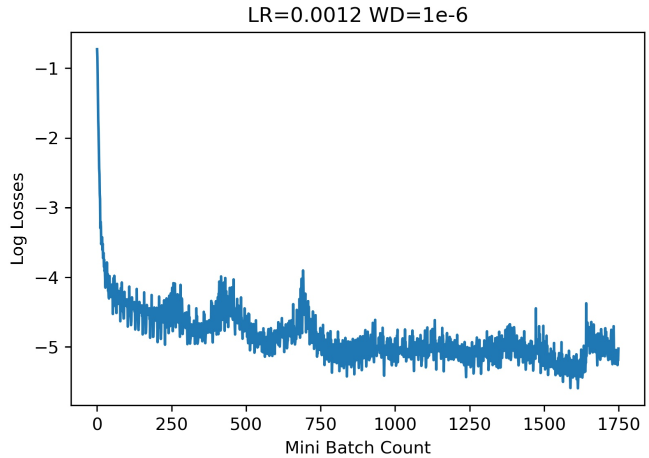

Figure 2: Log Training Loss

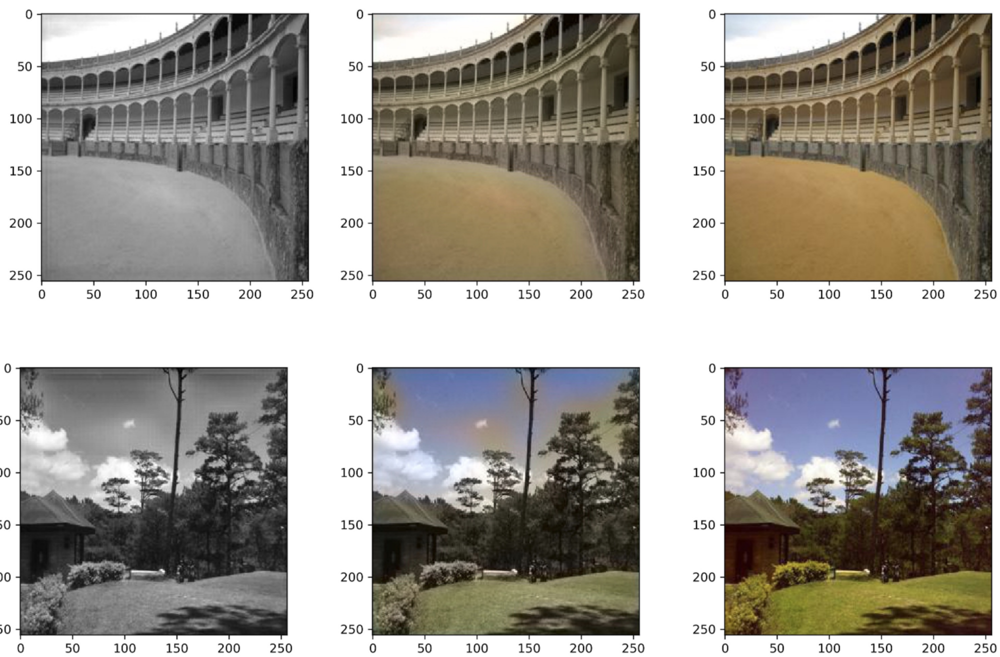

Figure 3: Training Images

training images colorized through the model to show the capabilities of the model and I also show test to present the pitfalls due to the small dataset.

4.3.1 Training Examples

In the below examples we can see that the model performs quite well on the training set. The left images are the input grayscale images, the middle are the model colorized images and the right are the ground truth images. Although the middle images do seem washed, as expected, their accuracy of colors and general look are quite good.

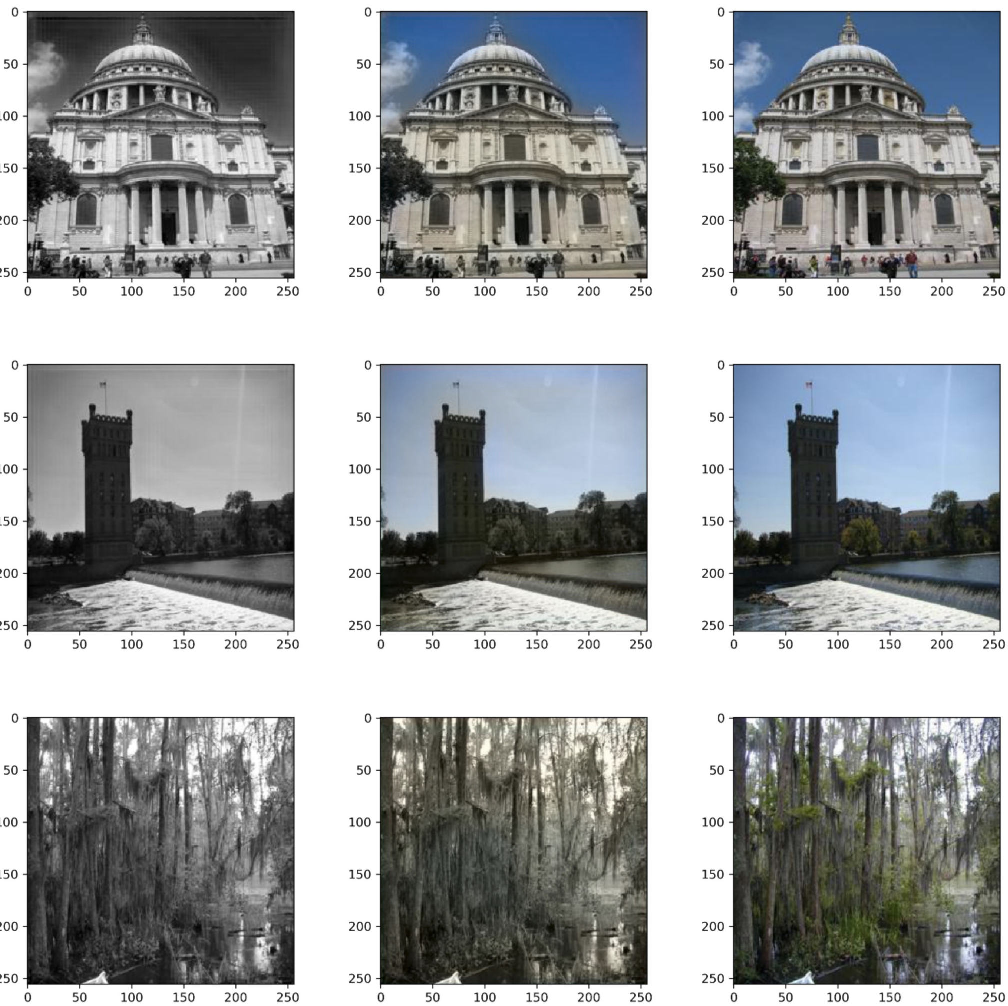

Figure 4: Test Images

4.3.2 Test Examples

The test images, as expected, do not generalize as well but that doesn't mean the performance isn't good. The image of the building on top is very good, most of my friends were not able to distinguish the difference. The model really falls apart in more complicated or unknown geometries such as the small leave of the swamp in the bottom image. The sky does not often change color in these images so it's not much of a surprise that the sky is rendered well in the tower image, but due to some images of the sunset we can see an orange wash that has to do with some of the averaging.

4.4 Improvements

Although I am generally happy with the performance of the model there are definitely areas of improvement. I will separate them into what is known to improve and what is unknown.

4.4.1 Known Improvements

Here are some improvements I would make if I were to continue working on the model.

- Larger Dataset: As mentioned prior, the small dataset plagues the model from being able to generalize well so an easy improvement would be to increase the dataset size at the cost of training time.
- Validation Loss: We cannot be sure if we are overfitting the training data if we don't keep track of the validation loss during training. With validation loss we could see if we overfit by seeing if the loss increases with more iteration and back the model up.
- More Training Epochs: Along with more data and the validation loss, we could run more epochs if needed.

4.4.2 Unknown Improvements

Here are some open issues that I am unsure how to resolve.

- Classification Loss: I had difficulty formulating the problem as a classification problem and it's been shown to improve the performance due to the color ambiguity.

References

(Fed17) Lucas Rodes-Guirao Federico Baldassarre Diego Gonzalez-Morin. "Deep-Koalarization: Image Colorization using CNNs and Inception-ResNet-v2". In: *ArXiv:1712.03400* (Dec. 2017). URL: `https://arxiv.org/abs/1712.03400`.

(ZIE16) Richard Zhang, Phillip Isola, and Alexei A. Efros. "Colorful Image Colorization". In: *CoRR* abs/1603.08511 (2016). arXiv: `1603.08511`. URL: `http://arxiv.org/abs/1603.08511`.

(Luo14) Ming Ronnier Luo. "CIELAB". In: *Encyclopedia of Color Science and Technology*. Ed. by Ronnier Luo. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 1–7. ISBN: 978-3-642-27851-8. DOI: `10.1007/978-3-642-27851-8_11-1`. URL: `https://doi.org/10.1007/978-3-642-27851-8_11-1`.

(Zho+17) Bolei Zhou et al. "Places: A 10 million Image Database for Scene Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017).